

APLIKASI PENJADWALAN MATA KULIAH MENGGUNAKAN ALGORITMA *PARTICLE SWARM OPTIMIZATION* (PSO)

Irfans Kusmarna¹, Luh Kesuma Wardhani², Muhammad Safrizal³

^{1,3}Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau

²Prodi Teknik Informatika, Fakultas Sains dan Teknologi, UIN Syarif Hidayatullah Jakarta

¹irfranstif@gmail.com, ²luhkesuma@uinjkt.ac.id, ³[safrizal.ilal@gmail.com](mailto:sufrizal.ilal@gmail.com)

ABSTRAK

Tujuan dari penelitian ini adalah untuk membangun aplikasi penjadwalan kuliah menggunakan algoritma *Particle Swarm Optimization* (PSO). Algoritma ini dipilih sebagai algoritma yang digunakan dalam penelitian ini karena algoritma PSO yang merupakan algoritma heuristik, adalah salah satu metode optimasi yang menggabungkan *local search* dan *global search*. Algoritma ini dapat memecahkan masalah dengan membentuk partikel-partikel pada populasi awal secara acak, mengevaluasi nilai *fitness*, dan meng-*update velocity* serta posisi dari partikel. Dengan demikian diharapkan output yang dihasilkan merupakan kombinasi terbaik dari serangkaian komponen yang ada. Kasus penjadwalan yang digunakan pada penelitian ini adalah penjadwalan mata kuliah di UIN Sultan Syarif Kasim Riau (UIN SUSKA). Permasalahan yang terjadi pada penjadwalan mata kuliah di UIN SUSKA Riau adalah tidak optimalnya pemenuhan beberapa *constraint* seperti keterbatasan ruangan. Dari hasil pengujian, aplikasi penjadwalan perkuliahan menggunakan algoritma PSO mampu menghasilkan jadwal perkuliahan yang sudah tidak ada bentrok tetapi masih tidak memenuhi dari segi kualitas yaitu jam dimulainya perkuliahan inti yang difokuskan pada jam-jam yang efektif.

Kata kunci: *Constraint, Fitness, Particle swarm optimizaton, Penjadwalan*

I. PENDAHULUAN

Penjadwalan merupakan proses, cara, pembagian waktu berdasarkan rencana pengaturan yang terperinci. Terdapat banyak hal yang harus dijadwalkan di suatu perguruan tinggi, di antaranya proses penerimaan mahasiswa baru, seminar, Ujian Tengah Semester (UTS), Ujian Akhir Semester (UAS), dan yang paling penting yaitu penjadwalan mata kuliah. Penjadwalan mata kuliah merupakan hal yang penting di dunia pendidikan. Penjadwalan yang baik dan memenuhi semua ketentuan universitas dan jurusan merupakan salah satu hal yang menjamin kelancaran pelaksanaan kegiatan belajar mengajar.

Permasalahan penjadwalan juga terjadi di UIN Sultan Syarif Kasim (Suska) Riau. Berdasarkan hasil wawancara yang dilakukan oleh peneliti dengan Sekretaris Jurusan Teknik Informatika UIN Suska Riau, terdapat beberapa aturan yang harus diperhatikan pada saat penjadwalan mata kuliah di Jurusan Teknik Informatika (TIF) UIN Suska Riau, yaitu:

- a) Setiap mata kuliah disajikan maksimal 2 (dua) kali sehari;
- b) Setiap dosen dijadwalkan mengajar 2 (dua) kali sehari;

- c) Tidak terdapat perkuliahan pada jam shalat Jumat;
- d) Tidak terdapat perkuliahan pada jam makan siang;
- e) Mata kuliah inti informatika dijadwalkan pagi hari;
- f) Dosen praktisi dijadwalkan mengajar hari sabtu;
- g) Dosen TIF dan non-TIF dijadwalkan mengajar pada hari senin-jumat.
- h) Mata kuliah pilihan dijadwalkan siang hari;
- i) Tidak ada perkuliahan pada hari Kamis dari pukul 08.00–12.00 bagi mahasiswa semester I, II, dan III;
- j) Waktu perkuliahan yang tersedia adalah hari Senin–Rabu adalah antara pukul 08.00–16.00 WIB, dan hari Kamis–Jumat antara pukul 08.00–16.30 WIB.

Untuk menyelesaikan masalah tersebut digunakan algoritma *Particle Swarm Optimization* (PSO). Algoritma ini dapat memecahkan masalah dengan membentuk partikel-partikel pada populasi awal secara acak, mengevaluasi nilai *fitness*, dan meng-*update velocity* serta posisi dari partikel berdasarkan ruang pencarian.

Diharapkan dengan bantuan *Particle Swarm Optimization* (PSO) penyusunan penjadwalan mata kuliah dapat dioptimalkan. Rumusan masalah dari penelitian ini adalah, “Bagaimana membangun aplikasi penjadwalan dengan menggunakan *Particle Swarm Optimization* (PSO) pada Jurusan Teknik Informatika

UIN Suska Riau". Sedangkan batasan masalah dalam penelitian ini adalah:

1. Mata kuliah yang akan mengalami proses dalam *Particle Swarm Optimization* (PSO) hanya mata kuliah yang sifatnya memerlukan ruangan, hari dan waktu tertentu.
2. *Constraint* yang digunakan bersifat statis yang diinputkan pada program.
3. *Constraint* mengikuti aturan yang berlaku pada penyusunan jadwal kuliah di jurusan teknik informatika UIN Suska Riau.

II. TINJAUAN PUSTAKA

2.1 Penjadwalan Kuliah

Menurut Putra (2009)[2], penjadwalan merupakan proses untuk menyusun suatu jadwal atau urutan proses yang diperlukan dalam sebuah persoalan. Persoalan penjadwalan biasanya berhubungan dengan penjadwalan kelas dalam sekolah atau perkuliahan dan juga dalam lingkup yang tidak jauh berbeda seperti penjadwalan ujian, penjadwalan karyawan, ataupun penjadwalan *job shop*. Dalam penjadwalan kuliah, akan dibahas tentang pembagian jadwal untuk tiap mahasiswa pada kuliah tertentu sekaligus dosen pengajarnya. Dalam penjadwalan ujian akan dibahas pengaturan dosen yang menjaga ujian dan mahasiswa yang menempati ruang ujian yang ada.

- a. Mendefinisikan atau membuat model dari permasalahan. Model yang dibuat mencakup proses apa saja yang akan dikerjakan dalam persoalan penjadwalan yang ada.
- b. Mendesain metode penyelesaian untuk permasalahan penjadwalan tersebut. Dari model yang telah ada, ditentukan metode yang digunakan untuk menyelesaikan permasalahan penjadwalan tersebut.
- c. Mencari bermacam-macam contoh permasalahan penjadwalan yang telah dibuat. Dalam proses ini dilakukan pencarian penyelesaian penjadwalan yang pernah digunakan agar dapat dipakai sebagai referensi dalam proses yang sedang dilakukan.

2.2 Particle Swarm Optimization (PSO)

PSO, merupakan algoritma berbasis populasi yang mengeksplorasi individu dalam pencarian. Dalam PSO populasi disebut *swarm* dan individu disebut *particle*. Tiap partikel berpindah dengan kecepatan yang diadaptasi dari daerah pencarian dan menyimpannya sebagai posisi terbaik yang pernah dicapai. Algoritma PSO ini awalnya diusulkan oleh J. Kennedy dan R. C. Eberhart.

PSO didasarkan pada perilaku sosial sekawanan burung atau sekumpulan ikan. Algoritma PSO meniru

perilaku sosial organisme ini. Perilaku sosial terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Kata partikel menunjukkan, misalnya, seekor burung dalam kawanan burung. Setiap individu atau partikel berperilaku secara terdistribusi dengan cara menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi perilaku kelompok kolektifnya. Dengan demikian, jika satu partikel atau seekor burung menemukan jalan yang tepat atau pendek menuju ke sumber makanan, sisa kelompok yang lain juga akan dapat segera mengikuti jalan tersebut meskipun lokasi mereka jauh di kelompok tersebut.

Algoritma PSO terdiri dari tiga tahap[1], yaitu pembangkitan posisi serta kecepatan partikel, *update velocity* (*update* kecepatan), *update position* (*update* posisi).

Pertama posisikan kecepatan dari kumpulan partikel dibangkitkan secara *random* menggunakan batas atas (x_{max}) dan batas bawah (x_{min}) dari *design variable*, seperti yang ditunjukkan pada persamaan (1) dan (2) (3):

$$x_0^i = x_{min} + rand(x_{max} - x_{min}) \quad (1)$$

$$v_0^i = x_{min} + rand(x_{max} - x_{min}) \quad (2)$$

Dimana:

x_0^i = posisi awal

v_0^i = kecepatan awal

x_{min} = batas bawah

x_{max} = batas atas

rand = nilai *random* antara nilai 0 dan 1

Dalam proses *random* diperlukan slot untuk menampung dari *random* posisi dan *random* kecepatan. Melalui proses pembangkitan partikel ini dapat maka kumpulan partikel dapat terdistribusi secara acak. Posisi dan kecepatan dari partikel direpresentasikan melalui vector.

$v_{k+1}^i v_k^i p_k^g$ Langkah kedua adalah *update velocity* (kecepatan) untuk semua partikel pada waktu $k+1$ menggunakan fungsi objektif atau nilai *fitness* [4] posisi partikel saat ini pada *design space* saat waktu ke k . Dari nilai *fitness* dapat ditentukan partikel mana yang memiliki nilai *global* terbaik (*global best*) pada *swarm* saat ini, dan juga dapat ditentukan posisi terbaik dari tiap partikel pada semua waktu yang sekarang dan sebelumnya (p^i). Perumusan *update velocity* menggunakan dua informasi tersebut untuk semua partikel pada kumpulan dengan pengaruh perpindahan yang sekarang untuk memberikan arah pencarian untuk generasi selanjutnya.

Perumusan *update velocity* mencakup beberapa parameter *random* (*rand*), untuk mendapatkan cakupan

yang baik pada *design space*, tiga parameter yang mempengaruhi arah pencarian, yaitu *inertia factor* (w), *self confidence* (c_1), *swarmconfidence* (c_2) akan digabungkan dalam satu penyajian, seperti yang ditunjukkan persamaan berikut:

$$v_{k+1}^i = w * v_k^i + c_1 * \text{rand} * (p^i - x_k^i) + c_2 * \text{rand} * (p_k^g - x_k^i) \quad (3)$$

Dengan range $w=0.4-0.9$, $\text{rand}=\text{interval } 0-1$,

Langkah terakhir dari setiap iterasi adalah *update* posisi tiap partikel dengan vektor *velocity*, seperti yang ditunjukkan pada persamaan berikut:

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (4)$$

Tiga tahapan di atas akan diulang sampai kriteria kekonvergenan terpenuhi, kriteria kekonvergenan sangat penting dalam menghindari penambahan fungsi evaluasi setelah solusi optimum didapatkan, namun kriteria kekonvergenan tidak selalu mutlak diperlukan, penetapan jumlah iterasi maksimal juga dapat digunakan sebagai *stopping condition* dari algoritma.

Banyak cara untuk membangun kondisi berhenti, di antaranya adalah: iterasi dihentikan ketika PSO telah mencapai iterasi maksimum, atau PSO telah menemukan nilai optimum tertentu atau kesalahan minimum yang diinginkan.

Algoritma PSO adalah sebagai berikut:

```

for setiap partikel
  Inialisasi partikel menggunakan persamaan (1) dan (2)
end
repeat
  for setiap partikel
    Hitung nilai fitness
    if nilai fitness baru lebih baik daripada nilai fitness
      lama
      Update nilai fitness dari partikel tersebut
  End
  Pilih partikel dengan nilai fitness terbaik diantara
  semua partikel tetangganya dan simpan nilai
  fitness terbaik tersebut
  for setiap partikel
    Hitung velocity partikel menggunakan
    persamaan (3)
    Update posisi partikel menggunakan
    persamaan (4)
  End
until (KriteriaBerhenti == true)

```

III. ANALISA DAN PERANCANGAN

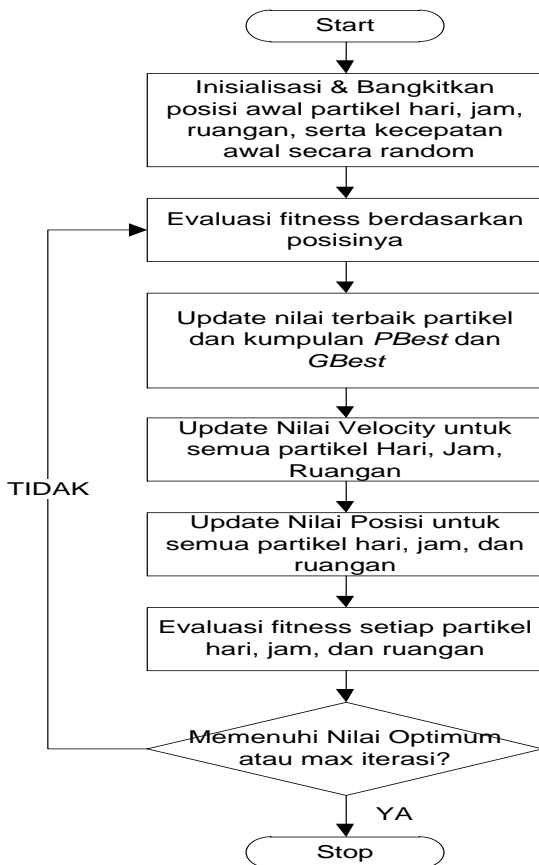
3.1 Aturan Penjadwalan Kuliah

Aturan-aturan penjadwalan di Jurusan Teknik Informatika UIN Suska Riau adalah sebagai berikut:

1. Ruang kelas tidak boleh dijadwalkan lebih dari satu perkuliahan secara bersamaan.
2. Perkuliahan mahasiswa tidak boleh dijadwalkan secara bersamaan.
3. Dosen tidak boleh dijadwalkan mengajar secara bersamaan.
4. Setiap mata kuliah disajikan maksimal 2 (dua) kali sehari.
5. Setiap dosen dijadwalkan mengajar 2 (dua) kali sehari.
6. Tidak terdapat perkuliahan pada jam shalat jumat.
7. Tidak terdapat perkuliahan pada jam makan siang.
8. Mata kuliah inti dijadwalkan pagi hari.
9. Dosen praktisi dijadwalkan mengajar hari Sabtu.
10. Dosen TIF dan non-TIF dijadwalkan mengajar pada hari senin-jumat.
11. Mata kuliah pilihan dijadwalkan siang hari.
12. Tidak ada perkuliahan pada hari Kamis dari pukul 08.00–12.00 bagi mahasiswa semester I, II, dan III.
13. Waktu perkuliahan yang tersedia adalah hari Senin–Rabu adalah antara pukul 08.00–16.00 WIB, dan hari Kamis–Jumat antara pukul 08.00–16.30 WIB.

3.2 Analisa PSO Pada Kasus Penjadwalan Mata Kuliah

Gambar 1 adalah diagram alir analisa PSO dalam penjadwalan mata kuliah di jurusan TIF UIN SUSKA Riau.



Gambar 1. Diagram alir analisa PSO dalam penjadwalan mata kuliah

Partikel yang diinisialisasi dalam penjadwalan mata kuliah ini meliputi data jam, ruang, hari, dosen, mata kuliah, dan ajar.

Tabel 1 Inisialisasi Data Jam

Id_Jam	Jam
1	08.00 – 08.50
2	08.50 – 09.40
3	09.40 – 10.30
4	10.30 – 11.20
5	11.20 – 12.10
6	13.00 – 13.50
7	13.50 – 14.40
8	14.40 – 15.30
9	15.30 – 16.20

Tabel 2 Inisialisasi Data Ruangan

Id_Ruangan	Nama Ruangan
1	TIF 301
2	TIF 302
3	TIF 303
4	TIF 304
5	TIF 305
6	PSI 101
7	PSI 102
8	PSI 103
9	LAB 1
10	LAB 2

Tabel 3. Contoh Pembangkitan posisi partikel pada iterasi pertama

Partikel 1				Partikel 2	Partikel 13	Partikel 4
Ajar				Hari	Jam	Ruangan
Id	MK	Dosen	Kelas			
1	1	26	A	1,1	8,1	3
2	1	26	B	2,6	4,6	4
3	1	26	C	2,1	4,6	4,5
4	1	25	D	2,8	6,4	5,2

Update *velocity* dan posisi menggunakan parameter uji coba yang diambil dari hasil penelitian yang dilakukan oleh Dian Ariani (Ariani, 2011) yang mana dengan parameter ini dapat menghasilkan rata-rata jadwal yang lebih optimal. Nilai tersebut adalah: $C1 = 1.5$; $C2 = 1.5$; $w = 0.5$.

Contoh hasil perhitungan update *velocity* dan posisi pada iterasi pertama dapat dilihat pada tabel 4.

Tabel 4. Contoh Hasil perhitungan *update velocity* dan posisi pada iterasi pertama

Partikel				R1	R2	P2	P3	P4
Ajar						Hari	Jam	Ruangan
Id	Makul	Dosen	Kelas					
1	1	26	A	0.3	0.2	1.7	12.2	4.5
2	1	26	B	0.3	0.4	3.9	6.9	6.0
3	1	26	C	0.1	0.2	3.2	4.9	7.8
4	1	25	D	0.1	0.3	4.2	9.6	7.8

Langkah selanjutnya adalah hitung kembali nilai *fitness*. Jika posisi semua partikel menuju ke satu nilai yang sama, maka ini disebut konvergen. Jika belum konvergen maka langkah 2 diulang dengan memperbaharui iterasi $i = i + 1$, dengan cara

menghitung nilai baru dari p^i dan p_k^g . Proses iterasi ini dilanjutkan sampai semua partikel menuju ke satu titik solusi yang sama. Biasanya akan ditentukan dengan kriteria penghentian (*stopping condition*), misalnya jumlah selisih solusi sekarang dengan solusi sebelumnya sudah sangat kecil.

3.3 Fungsi

Fungsi-fungsi yang dibutuhkan oleh sistem adalah sebagai berikut:

1. Fungsi Pembangkitan Posisi dan Kecepatan

Fungsi ini digunakan untuk inialisasi dan pembuatan populasi awal dengan mengacak semua data ajar, data ruangan, data hari, dan data jam menjadi partikel-partikel

2. Fungsi *Fitness*

Fungsi yang digunakan untuk menyatakan seberapa baik nilai dari suatu individu ataupun solusi yang didapatkan.

3. Fungsi *Update* Kecepatan

Fungsi ini digunakan untuk memperbarui nilai kecepatan berdasarkan nilai fungsi yang diberikan.

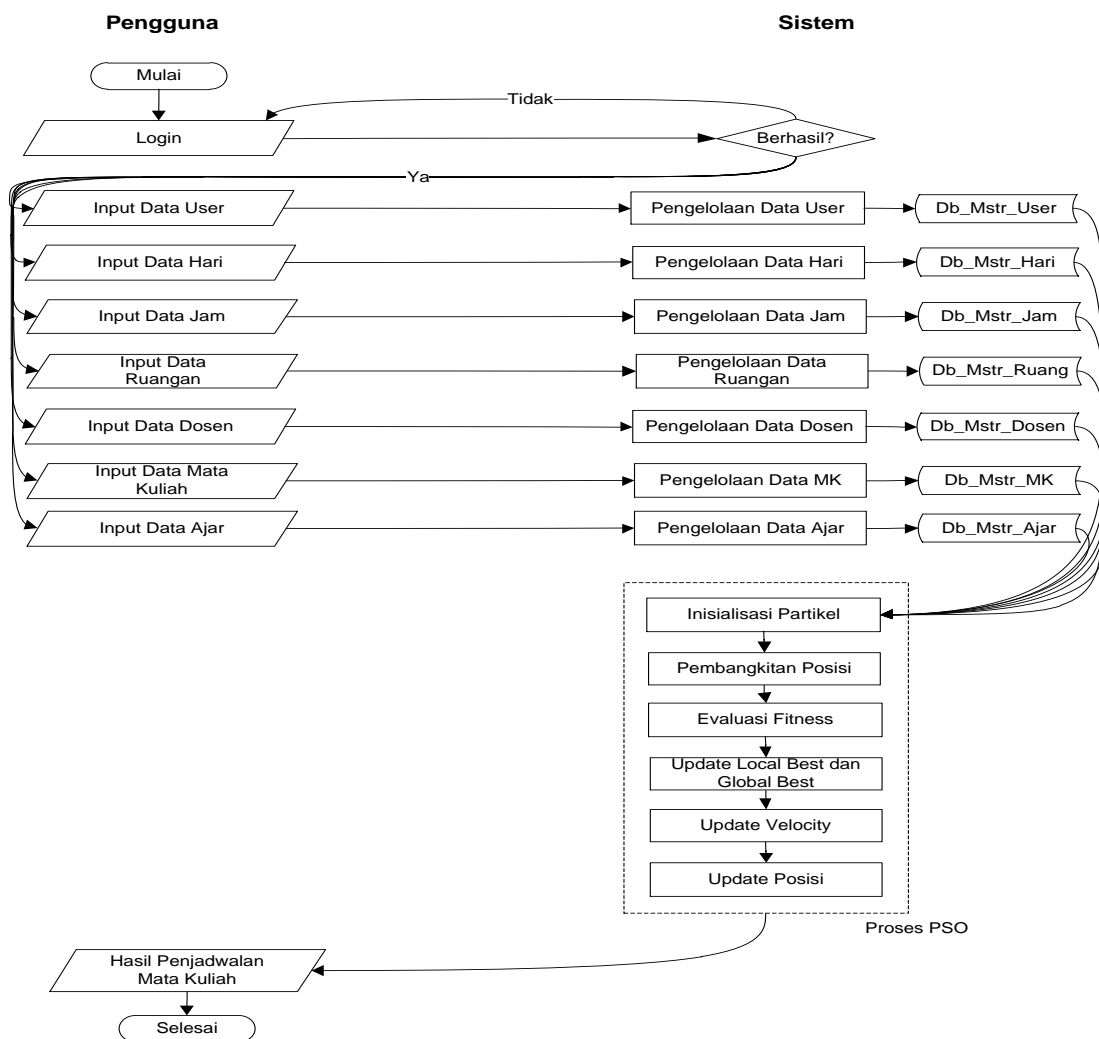
4. Fungsi *Update* Posisi

Fungsi ini digunakan untuk memperbarui nilai posisi baru partikel berdasarkan nilai kecepatan perpindahan partikel.

5. Fungsi *Setting* (Pengaturan)

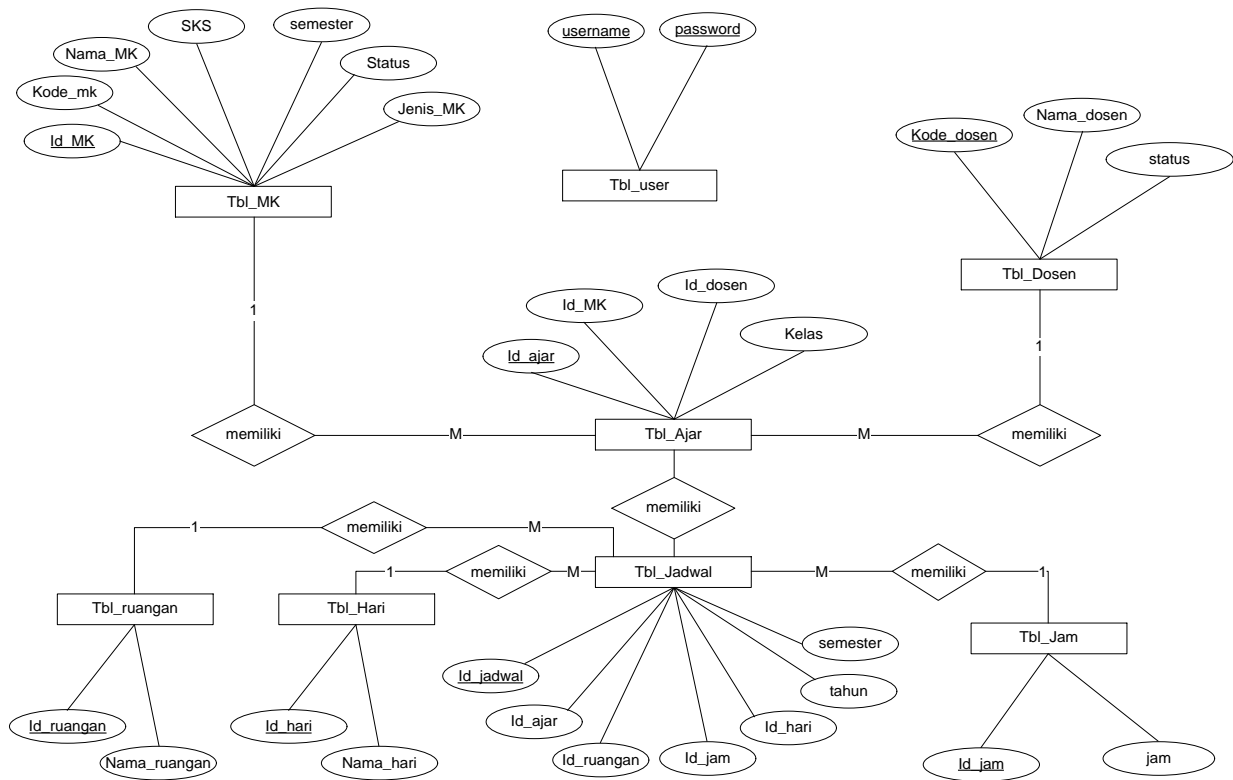
Dalam fungsi ini terdapat fasilitas untuk menentukan parameter-parameter dalam PSO, seperti nilai faktor *inertia*, *learning rates* (*self confidence*, *swarmconfidence*). Akan tetapi parameter ini juga memiliki nilai *default* untuk mengantisipasi bila perubahan nilai parameter menghasilkan kinerja yang kurang memuaskan.

3.3 Diagram Alir Sistem



Gambar 2. Diagram Alir Sistem

3.3 Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram (ERD)

IV. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tampilan menu utamadapat diakses jika menu *login* dinyatakan *valid*.



Gambar 5. Tampilan menu utama

Untuk menjalankan aplikasi ini, setelah diinputkan data masternya, proses selanjutnya adalah menekan *button* Penjadwalan, kemudian isi nilai parameter yang dibutuhkan atau bisa menggunakan nilai *default*.

Proses Penjadwalan

Tahun Ajaran	:	2011
Semester	:	Genap
C1	:	1.5
C2	:	1.5
W	:	0.5
Proses		

Gambar 6. Proses Pembuatan Jadwal

Setelah menekan tombol “proses”, sistem akan memulai proses pembuatan jadwal, nilai setiap iterasi akan disimpan dalam memori sementara. Setelah selesai tekan tombol “Lihat Hasil” untuk melihat data jadwal yang telah dibuat oleh sistem dengan menggunakan PSO.

Jadwal Kuliah Semester Genap Tahun 2010/2011
 JURUSAN TEKNIK INFORMATIKA
 FAKULTAS SAINS DAN TEKNOLOGI UIN SUSKA RIAU

Gambar 7. Laporan jadwal

4.2 Pengujian

Pengujian yang dilakukan pada aplikasi ini adalah:

1. Menggunakan *Black Box*

Hasil pengujian sistem yang dilakukan dengan menggunakan *black box* adalah:

Tabel 1. Hasil pengujian *black box*

No. Uji	Butir Uji	Hasil Pengujian
Uji 1	Pengujian Modul Login	Diterima
Uji 2	Pengujian Modul Menu <i>Input Data Hari</i>	Diterima
Uji 3	Pengujian Modul Menu <i>Input Data Jam</i>	Diterima
Uji 4	Pengujian Modul Menu <i>Input Data Ruangan</i>	Diterima
Uji 5	Pengujian Modul Menu <i>Input Data Dosen</i>	Diterima
Uji 6	Pengujian Modul Menu <i>Input Data Mata Kuliah</i>	Diterima
Uji 7	Pengujian Modul Menu <i>Input Data Ajar</i>	Diterima
Uji 8	Pengujian Modul Menu Proses	Diterima

Berdasarkan pengujian menggunakan *Black box*, seluruh menu dan *button* pada sistem penjadwalan perkuliahan ini berfungsi dengan baik.

2. Menggunakan Uji Performansi

Pengujian performansi yang dilakukan dengan pengujian menggunakan parameter nilai *default* dan parameter inputan *user*.

Tabel 2. Parameter Perhitungan dengan Nilai Default

C1	C2	w	Iterasi Maksimum
1,5	1,5	0,5	1000

Tabel 3. Pengujian Performansi dengan Parameter Nilai Default

Pengujian ke-	Jumlah Iterasi	Hasil Pengujian
1	40	Berhasil
2	28	Berhasil
3	32	Berhasil
4	39	Berhasil
5	29	Berhasil
6	37	Berhasil
7	23	Berhasil
8	29	Berhasil
9	45	Berhasil
10	23	Berhasil

Tabel 4. Pengujian Performansi dengan Parameter Input dari User

Pengujian ke-	C1	C2	w	Iterasi	Hasil Pengujian
1	1	2	0,6	38	Berhasil
2	3	2	0,6	48	Berhasil
3	0,7	1,4	0,4	40	Berhasil
4	3	1	0,5	28	Berhasil
5	2	0,5	0,9	34	Berhasil
6	2	1	0,8	41	Berhasil
7	4	1	0,5	32	Berhasil
8	1	4	0,5	37	Berhasil
9	5	1	0,4	31	Berhasil
10	1	7	0,7	42	Berhasil

Berdasarkan pengujian performansi, baik yang menggunakan parameter nilai *default* maupun yang menggunakan nilai inputan dari *user*, sistem ini

berhasil menghasilkan solusi penjadwalan mata kuliah yang sudah tidak terdapat bentrok terhadap hari, jam, dan ruangan perkuliahan.

3. Menggunakan *User Acceptance Test*

User Acceptance Test merupakan pengujian yang dilakukan dengan meminta persetujuan dari *user* terhadap *output* yang dihasilkan oleh aplikasi penjadwalan mata kuliah ini. Responden yang melakukan pengujian yaitu Ketua Jurusan atau Sekretaris Jurusan Teknik Informatika UIN Suska Riau. Hasil pengujian ini adalah: aplikasi ini berhasil menghasilkan jadwal perkuliahan yang sudah tidak terdapat bentrok, namun tidak berhasil memenuhi dari segi kualitas yaitu jam dimulainya perkuliahan yang difokuskan pada jam-jam yang efektif.

V. PENUTUP

5.1 Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Aplikasi penjadwalan mata kuliah menggunakan *Particle Swarm Optimization* (PSO) berhasil dirancang dan dibangun untuk menghasilkan jadwal perkuliahan di Jurusan Teknik Informatika UIN Suska Riau walaupun tidak semua *constraint* yang ditetapkan oleh pihak Jurusan terpenuhi.
2. Walaupun berhasil menghasilkan jadwal perkuliahan yang sudah tidak terdapat bentrok, namun tidak berhasil memenuhi dari segi kualitas yaitu jam dimulainya perkuliahan yang difokuskan pada jam-jam yang efektif.

3. Aplikasi penjadwalan mata kuliah ini mampu menangani proses input data, melakukan proses pembuatan jadwal perkuliahan secara otomatis dan menghasilkan jadwal yang dapat ditampilkan dan dapat pula dicetak.
4. Aplikasi penjadwalan mata kuliah menggunakan *Particle Swarm Optimization* (PSO) memiliki kekurangan yaitu *constraint* yang tidak bisa berubah sesuai kondisi.

5.2 Saran

Dari hasil pembahasan yang telah dilakukan, dapat disarankan untuk penelitian selanjutnya agar:

1. Penelitian mengenai performansi algoritma masih sangat dibutuhkan lebih lanjut pada bidang aplikasi lainnya, sehingga mampu memberikan kontribusi pada perkembangan algoritma tersebut.
2. Untuk pengembangan selanjutnya, *constraint* dibuat dapat diubah sesuai kebutuhan dan persyaratan pembuatan jadwal yang berlaku.
3. Jam dimulainya perkuliahan difokuskan pada jam-jam yang efektif.

DAFTAR PUSTAKA

- [1] Ariani, Dian, 2011, *Optimasi Penjadwalan Mata Kuliah di Jurusan Teknik Informatika PENS Dengan Menggunakan Algoritma Particle Swarm Optimization* (PSO), Surabaya, Institut Teknologi Sepuluh Nopember.
- [2] Putra, Yendrika, 2009, *Aplikasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetika*, Pekanbaru UIN Suska Riau
- [3] Santosa, Budi, dan Paul Willy, 2011, *Metoda Metaheuristik Konsep dan Implementasi*, Surabaya, Guna Widya.
- [4] Suyanto, 2008, *Evolutionary Computation*, Bandung, Informatika.